

Tentamen i Programmering i R, 7.5 hp

Skrivtid: 8.00-12.00

Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" och några andra referenskort tillgängliga elektroniskt.

Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.

Tänk på följande:

Skriv dina lösningar i **fullständig och läsbar kod**.

Lösningen skrivs i en körbar R-fil med namnet **tentaXX.R** där XX är ditt tenta-ID

Tex: tenta01.R om ditt tenta-ID är 01. Lämna *inte* in något Word-dokument!

Se filen **DocStudent.pdf** för hur tentan ska lämnas in.

Kommentera direkt i R-filen när något behöver förklaras eller diskuteras.

Eventuella grafer som skapas under tentans gång behöver **INTE** skickas in för rättning, det räcker med att **skicka in den kod som producerar figurerna**.

OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.

1. Datastrukturer och beräkningar (4p)

- (a) Gör följande beräkning $(a! - b^a)^{-0.1}$, där $a = 5$ och $b = 2$. ! indikerar fakultet. **1p**
- (b) Utgå från det inbyggda datasetet **OrchardSprays** och skapa en data.frame som du kallar **my_df**. Den ska innehålla de två första kolumnerna och alla rader med udda indexnummer (givet den ursprungliga sorteringen). **1p**
- (c) Skapa en textvektor som du kallar **my_text**. Elementen ska vara ordnade på så sätt att sekevensen a, b, c, \dots, x, y, z (hela det engelska alfabetet) ska upprepas 1000 gånger. **1p**
- (d) Återskapa matrisen nedan. **1p**

	[,1]	[,2]	[,3]
w	1	1	1
a	2	2	2
r	3	3	3
y	2	2	2
f	1	1	1

2. Kontrollstrukturer (4p)

- (a) Använd en for-loop med 7 iterationer för att skapa en konsolutskrift enligt nedan. Varje iteration ska skriva ut minst en rad. **2p**

```
[1] "xyz" "xyz" "xyz" "xyz" "xyz" "xyz" "xyz"
[1] "abc"
[1] "xyz" "xyz" "xyz" "xyz" "xyz" "xyz"
[1] "abc" "abc"
[1] "xyz" "xyz" "xyz" "xyz" "xyz"
[1] "abc" "abc" "abc"
[1] "xyz" "xyz" "xyz" "xyz"
[1] "abc" "abc" "abc" "abc"
[1] "xyz" "xyz" "xyz"
[1] "abc" "abc" "abc" "abc" "abc"
[1] "xyz" "xyz"
[1] "abc" "abc" "abc" "abc" "abc" "abc" "abc"
[1] "xyz"
[1] "abc" "abc" "abc" "abc" "abc" "abc" "abc"
```

- (b) Skapa en while-loop som gör följande: Beräknar den kumulativa summan för talen 1, 2, 3, 4, ...
I varje iteration ska följande skrivas ut:
- Om summan är jämt delbar med 2: skriv ut "Go!"
 - Om summan är jämt delbar med 5 men inte jämt delbar med 2: skriv ut sinus av summan
 - Annars: Skriv ut summans värde.
- Loppen ska avbrytas om summan är större än 50. Om du gjort rätt ska följande utskrift erhållas: **2p**

```
[1] 1
[1] 3
[1] "Go!"
[1] "Go!"
[1] 0.650288
[1] 21
[1] "Go!"
[1] "Go!"
[1] 0.850904
```

3. Strängar och datum (4p)

- (a) Läs in paketen `lubridate` och `stringr` i R. Läs is textfilen "wiki_robot.txt" till R och spara som en vektor som du kallar `robot`. **0.5p**

- (b) Använd funktioner ur **stringr** för att räkna ut hur många gånger ordet “The”/“the” förekommer totalt i **robot**. Andra ord som innehåller bokstavskombinationen “the” ska inte räknas, t.ex. “there”. **1.5p**
- (c) 24 maj 1919 infördes allmän rösträtt i Sverige, vilket innebar att kvinnor fick möjlighet att rösta. Det första valet efter detta beslut var Andrakammarvalet i Sverige 1921, som hölls den 12 september 1921. Polen införde allmän rösträtt 28 November 1918. Använd funktioner i R för att svara på frågorna nedan. **2p**
- Vilken veckodag infördes allmän rösträtt i Sverige?
 - Hur många dagar var det mellan infördandet av allmän rösträtt i Sverige och Andrakammarvalet i Sverige 1921? Inkludera gränserna.
 - Hur många hela veckor var det mellan att Polen införde allmän rösträtt och Andrakammarvalet i Sverige 1921? Inkludera gränserna.
 - Hur många hela månader är det mellan infördes allmän rösträtt i Sverige och dagens datum? (2017-08-14) Inkludera gränserna.

4. Funktioner: (4p)

- (a) Nu ska du skapa en funktion **my_curve(x, a=3)** som implemeterar den matematiska funktionen nedan för en numerisk vektor x. Se testfallen för hur funktionen ska fungera. **2p**

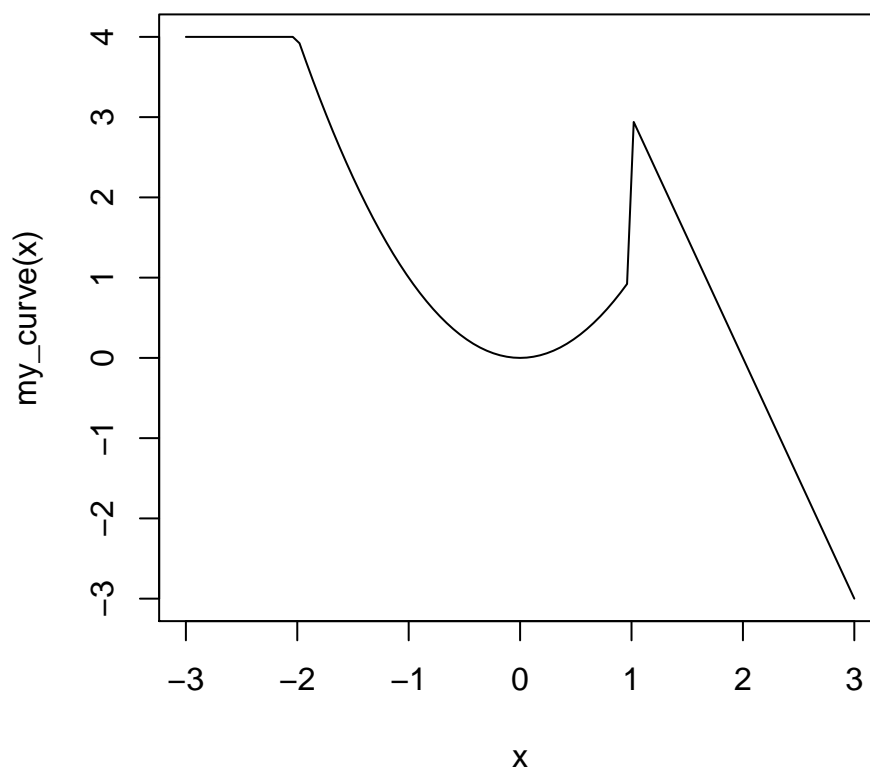
$$f(x) = \begin{cases} 4 & x \leq -2 \\ x^2 & -2 < x < 1 \\ 6 - a \cdot x & 1 \leq x \end{cases}$$

```
my_curve(x = c(-10,-20,-2,-1.5))
[1] 4.00 4.00 4.00 2.25

my_curve(x = c(-2,0,0.99,1))
[1] 4.0000 0.0000 0.9801 3.0000

my_curve(x = c(-1,0,1,2,4.5), a = 5)
[1] 1.0 0.0 1.0 -4.0 -16.5

curve(expr = my_curve, from = -3, to = 3)
```



- (b) Nu ska du skapa en funktion som ska beräkna variansen för en numerisk vektor. Kalla funktionen `my_var(x)`. Inga inbyggda funktioner som beräknar varians, standardavvikelse eller medelvärde får användas. Beräkna variansen enligt formeln:

$$V = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Där x_i är elementen i vektorn \mathbf{x} . n är antal element i vektorn. \bar{x} är medelvärdet av \mathbf{x} . Se exemplen nedan för hur funktionen ska fungera. **2p**

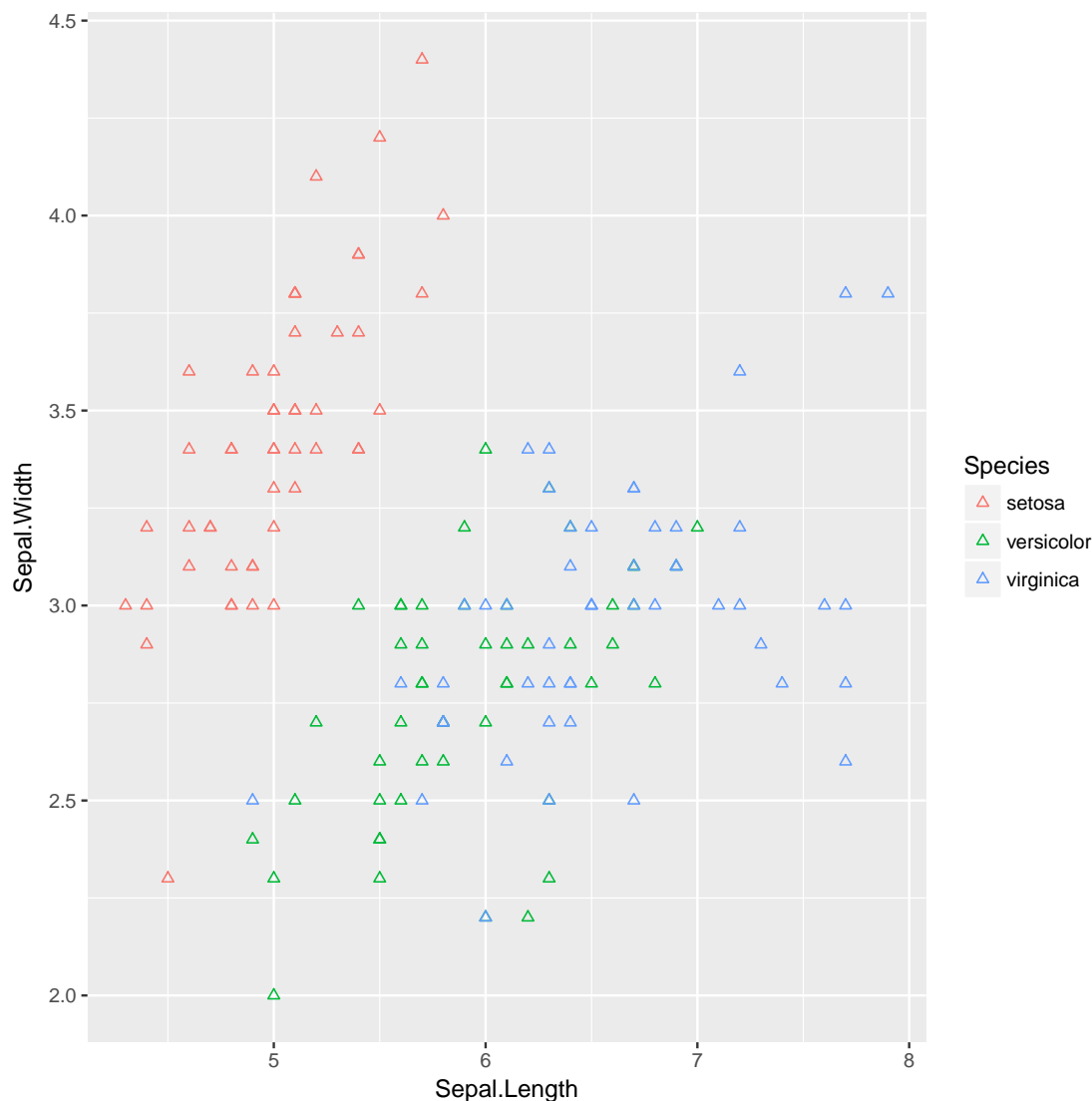
```
my_var(x = 1:100)
[1] 841.667

my_var(x = c(2,4,7,2,2.5,6))
[1] 4.64167

my_var(x = c(-3,4,6,5,-10))
[1] 46.3
```

5. Statistik och grafik (4p)

- (a) Utgå från det inbyggda datasetet `iris` och använd funktioner från `ggplot2` för att återskapa figuren nedan. **2p**



- (b) Använd `iris`-data: Gör ett *two-sample t-test* där ni testar om medelvärdet för *Petal.Length* är mindre för arten *versicolor* än för arten *virginica*. Använd signifikansnivå 0.1. Spara t-värdet i variabeln `t_value` och p-värdet i variabeln `p_value`. Anta att varianserna mellan grupperna är olika. **1.5p**
- (c) Fortsätt med `iris`-data. Ta reda på vilken art som har minst median på variabeln *Sepal.Width*. **0.5p**

Kom ihåg: Lösningen skrivs i en körbar R-fil med namnet `tentaXX.R` där XX är ditt tenta-ID tex: `tenta01.R` om du har tenta-ID är 01. Lämna *inte* in något Word-dokument!

Lycka till!